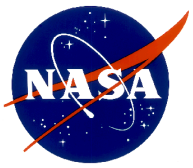
JPL Publication 09-6



Analysis of Single-Event Upset Rates in Triple-Modular Redundancy Devices

Larry D. Edmonds

National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

February 2009

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Abstract

Devices hardened against single-event upset (SEU) via triple-modular redundancy (TMR) require unconventional methods for estimating system error rates in a space environment. We cannot simply integrate a cross section versus linear energy transfer (LET) curve with an environmental LET spectrum because "cross section" has a flux dependence. The strategy proposed here is to experimentally measure (at a particle accelerator) system error rates as a function of "raw" (i.e., with TMR disabled) bit-flip rates. The same experiment also measures cross section versus LET for raw bit flips, and this information is used to calculate the raw rate in a space environment. The calculated raw rate together with the measured system rate versus raw rate curve provides an estimate of the system rate. Theoretical predictions of system rate versus raw rate are provided here so that data obtained at flux levels practical for experimental work can be extrapolated to the lower flux levels encountered in space.

Contents

Figures

1. Introduction

Various types of error detection and correction (EDAC) mechanisms have been used to mitigate the impact of single-event upset (SEU) bit errors. The function of EDAC in general is such that bit errors are normally not observable on a system level. However, the current forms of EDAC have limitations, and bit errors can become observable under certain conditions. Different types of EDAC differ in terms of what those conditions are. One type uses the Hamming code. Another type is the subject of this report and is based on triple-modular redundancy (TMR). Section 2 states the conditions that define a system error, which in turn defines the TMR problem considered here. Section 3 derives the probability of a system error during a single cycle, and Section 4 uses this result to derive an equation that calculates the system error rate. Section 5 derives an alternate form of the rate equation, which expresses the rate in terms of various moments associated with a set of numbers of bits. This is useful for extracting information regarding device architecture from accelerator test data, with the information consisting of estimates of the moments. Section 6 shows how these moments can be used to quickly and easily compare different architectures to determine which has the larger error rate. A method for extracting information from accelerator test data is given in Section 7, which also includes examples.

2. Definition of the Problem Considered

The problem considered is defined by defining a system error, but it is necessary to first describe the assumed device architecture. The device is partitioned into some number M of groups of bits. Each group, in turn, is partitioned into three modules. Each module within a common group contains the same number of bits, but this number can be different for different groups. The bit partitioning is shown in Figure 1, which describes the logical bit mapping although the physical locations of the bits may not be as shown. Each block in the figure is called a block here and is identified by both a group and a module. The operational time is also partitioned. It is partitioned into a set of time windows called cycles here. During any given cycle, single-event bit errors can occur anywhere within the device. However, bit errors created during one cycle are not present during the next cycle because one of the functions of the TMR is to correct these errors from one cycle to the next; unless a system error (defined below) occurs, in which case the resetting is done by external intervention. Therefore, we can consider individual cycles as independent entities. For grammatical brevity, we will say that two or more bit upsets are "simultaneous" if they were produced during the same cycle. One or more simultaneous bit errors within a given block will be called a block error. The function of the TMR is such that a block error is not relevant (does not produce a system error) if it is not simultaneous with any other block error. An event that is relevant, and will be called a group error, consists of simultaneous errors in two (or three) distinct blocks within the same group (i.e., distinct modules). For any given cycle, a system error is defined to be the occurrence of one or more group errors. For example, one or more bit errors in Block (1,3) (see Fig. 1) is not a system error by itself, but a system error occurs if this is accompanied by simultaneous bit errors (one or more) in Block (1,1). This is one example of a Group 1 error and is also an example of a system error. Similarly, a system error occurs if an error in Block (2,1) is simultaneous with an error in Block (2,2). This is one example of a Group 2 error and is a second example of a system error. A

third example of a system error is that in which both of the above events occur during the same cycle. Note that this third example counts as one system error, i.e., there can be at most one system error per cycle. The objective is to calculate the system error rate.

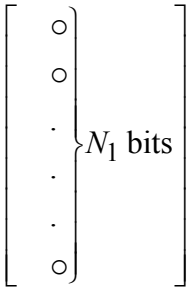
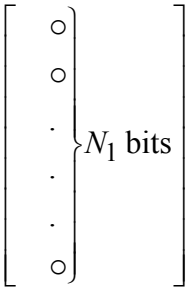
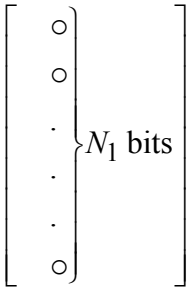
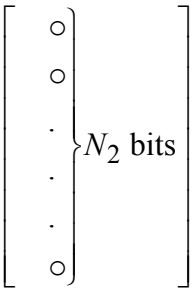
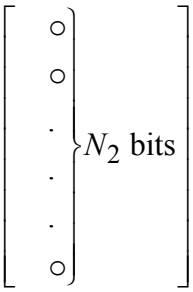
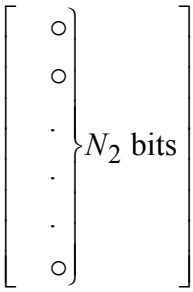
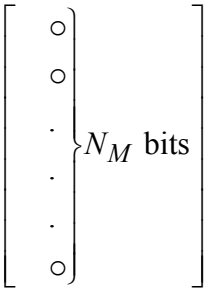
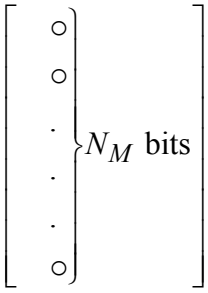
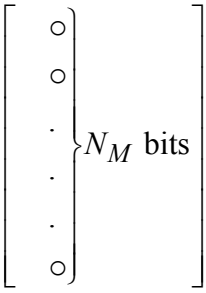
	Module 1	**Module 2**	**Module 3**
Group 1	 Block (1,1)	 Block (1,2)	 Block (1,3)
Group 2	 Block (2,1)	 Block (2,2)	 Block (2,3)
.	.	.	.
Group M	 Block (M,1)	 Block (M,2)	 Block (M,3)

Figure 1. Bit Partitioning.

The bits are partitioned into M groups and three modules that intersect to form blocks. This bit partitioning describes the logical bit mapping, but the physical locations of the bits may not be as shown. The function of TMR is such that one or more bit errors in the same block is a block error, two or more block errors in the same group is a group error, and one or more group errors is a system error.

A good device layout ensures that multiple bit upsets produced by a single ion hit will not produce multiple block errors within a common group, and such a layout is assumed in the analysis that follows. A system error then requires that two or more ions each hit a different block within a common group.

3. Expected Number of System Errors During a Selected Cycle

Rates calculated in the next section will use results derived in this section. This section calculates the expected (statistical average) number of system errors during one arbitrary but fixed cycle. Because only one cycle is considered, the term "simultaneous" is not needed because all bit errors considered are automatically simultaneous. Note that the number of system errors is either zero or one, so the expected number of system errors is equal to the probability of a system error. This probability is calculated from three steps. The first step calculates the probability of a block error. The second step expresses the probability of a group error in terms of the probability of a block error, and the third step expresses the probability of a system error in terms of the probability of a group error.

Before starting the first step, we define notation that will be used in all of the steps. An outcome is the specification of the error state (yes or no) of every bit in the device. The outcome set $E_{i,j}$ (where $i = 1,\ldots,M$ and j=1,2,3) is the set of all outcomes in which there was a block error in the (i,j) block, i.e., one or more bit errors in the (i,j) block. The outcome set E_i (where $i = 1,\ldots,M$) is the set of all outcomes in which there was a group error in the i^{th} group. Note that such an event occurs if (and only if) there is either: a block error in the (i,1) block and in the (i,2) block, or a block error in the(i,1) block and in the (i,3) block, or a block error in the(i,2) block and in the (i,3) block. Using elementary set theory notation, this is written as

$$E_i = \left[E_{i,1} \cap E_{i,2}\right] \cup \left[E_{i,1} \cap E_{i,3}\right] \cup \left[E_{i,2} \cap E_{i,3}\right]. \tag{1}$$

Note that outcomes producing more than one of the above possibilities (i.e., that produce errors in all three blocks in the i^{th} group) are also included in (1). The outcome set E is the set of all outcomes that produce a system error. Note that such an event occurs if (and only if) there is a group error in at least one group so

$$E = \bigcup_{i=1}^{M} E_i \ . \tag{2}$$

We now calculate the probability of a block error. Consider the (i,j) block, which contains N_i bits. The probability of an error in this block is denoted $P(E_{i,j})$ and is the probability of one or more bit errors in a group of N_i bits. This probability is calculated from Poisson statistics using

$$P(E_{i,j}) = 1 - \exp(-N_i \, r \, T_C) \tag{3}$$

where T_C is the cycle duration and r is the expected per-bit upset rate and is assumed to be the same for all bits. The rate r is calculated from a measured per-bit cross section together with an environmental model representing the environment of interest.

We next calculate the probability of a group error in the i^{th} group, which is denoted $P(E_i)$. This calculation is complicated by the fact that the square brackets on the right side of (1) are not

mutually exclusive outcome sets, and they are also not statistically independent. For illustration, suppose it is given that an outcome is in the first square bracket. Then it is also in the set $E_{i,1}$, which increases the probability (compared to what the probability would be if the stated assertion were not given) that this outcome will also be in the second square bracket, hence, the two square brackets are not statistically independent. However, the three sets ($E_{i,1}$, $E_{i,2}$, and $E_{i,3}$) are statistically independent because they refer to bit errors in distinct bit blocks that have no bits in common. An equation that takes advantage of this fact can be derived by noting that the probability function is an additive set function that satisfies

$$P(A \bigcup B) = P(A) + P(B) - P(A \bigcap B)$$

for any two outcome sets A and B. A second application of this equation gives

$$P(A \bigcup B \bigcup C) = P(A) + P(B) + P(C) - P(A \bigcap B) - P(A \bigcap C) - P(B \bigcap C) + P(A \bigcap B \bigcap C) . \quad (4)$$

If we now let A, B, and C be the three square brackets in (1) we obtain

$$P(E_i) = P(E_{i,1} \bigcap E_{i,2}) + P(E_{i,1} \bigcap E_{i,3}) + P(E_{i,2} \bigcap E_{i,3}) - 2P(E_{i,1} \bigcap E_{i,2} \bigcap E_{i,3}) . \quad (5)$$

The sets $E_{i,1}$, $E_{i,2}$, and $E_{i,3}$ are statistically independent so probabilities of intersections are products of probabilities. Also, (3) gives $P(E_{i,1})=P(E_{i,2})=P(E_{i,3})$ so (5) becomes

$$P(E_i) = 3P^2(E_{i,1}) - 2P^3(E_{i,1}) . \quad (6)$$

We finally calculate the probability $P(E)$ of a system error. Note that the sets inside the union in (2) are statistically independent because they refer to bit errors in distinct bit groups that have no bits in common. An equation that takes advantage of this fact can be derived by noting that the probability function satisfies

$$P(A) = 1 - P(A^*)$$

where A is any outcome set and * denotes the compliment, i.e., A^* is "not A". We also have

$$E^* = \left(\bigcup_{i=1}^{M} E_i \right)^* = \bigcap_{i=1}^{M} E_i^* .$$

Using these results together with the fact that the sets E_1^*, E_2^*,…, E_M^* are statistically independent, so probabilities of intersections are products of probabilities, gives

$$P(E) = 1 - P(E^*) = 1 - P\left(\bigcap_{i=1}^{M} E_i^* \right) = 1 - \prod_{i=1}^{M} P(E_i^*) = 1 - \prod_{i=1}^{M} [1 - P(E_i)] . \quad (7)$$

Combining (3) with (6) and (7) gives

$$P(E) = 1 - \prod_{i=1}^{M} [3\exp(-2N_i \, r \, T_C) - 2\exp(-3N_i \, r \, T_C)] . \quad (8)$$

4. Rates and Accumulated Numbers of Errors

The previous section derived the expected number of system errors during a single cycle. We now consider the expected number accumulated over a time period that contains multiple cycles. "Accumulated number" is defined to mean that the expected number from one cycle is added to that from another cycle. Therefore, we calculate the accumulated number over a given time period by simply summing the numbers over all cycles contained in that time period. One way to perform this summation is by taking the time integral of the system error rate. This rate is the expected number of system errors from one cycle divided by the duration of the cycle T_C. The expected number is the same as the probability given by (8), so the system error rate, denoted R, is given by

$$R = \frac{1}{T_C} - \frac{1}{T_C}\prod_{i=1}^{M}[3\exp(-2N_i\, r\, T_C) - 2\exp(-3N_i\, r\, T_C)]. \tag{9}$$

If the environment changes with time, so the bit error rate r changes with time, then the system error rate R will have a time dependence and is integrated in time to obtain the expected number of system errors over a given time period. If R is constant, we simply multiply by time to obtain the number of errors. Strictly speaking, R is not an instantaneous rate. It is the rate averaged over the cycle time T_C. If this cycle time is long enough for the environment to change significantly during a cycle, then the bit rate r that should be used in (9) is the bit rate averaged over the cycle time. If the cycle time is short compared to other time scales of interest (including time scales associated with environmental changes), then r and R can both be regarded as instantaneous rates.

5. Approximations

This section derives two approximations for (9), with one (the simplest) valid when r is sufficiently small, and the other (slightly more complex) valid for arbitrary r. Note that (9) is exact for arbitrary r, so it is reasonable to ask why approximations are needed. There are three reasons:

(a) *Required Numerical Precision*: When r is small, the right side of (9) subtracts nearly equal numbers, which is a numerical problem. The exact equation when evaluated with finite arithmetic precision can actually be less accurate than an approximation evaluated with the same arithmetic precision.

(b) *Simplicity Provides Insight*: To determine the exact dependence that R has on r, it is necessary to assign numerical values to each of the parameters N_1, N_2, …, N_M. However, the approximations derived below will show that this dependence is most strongly controlled by just a few "moments" associated with these parameters (two examples of moments are the mean and the root-mean-square). Simple approximations containing a few moments provide insight that can easily answer questions whose answers are not at all obvious from a casual inspection of (9). Examples of such questions are given in the next section.

(c) *Curve Fitting*: In some practical applications, the parameters N_1, N_2, …, N_M are not known inputs. Instead, the available information consists of measurements of R as a function of r, performed at a particle accelerator in which errors are counted at different flux levels, with each flux level producing a different value for r (i.e., each flux level produces one data point in a plot of R versus r). Curve fitting is needed to extract parameters from such data. Unfortunately, this method is not able to extract a large number of parameters from a smaller number of data points, even in a hypothetical experiment that produces no experimental scatter. However, by recognizing that the dependence of R on r is most strongly controlled by a few moments, discussed in (b) above, and by using these moments as the fitting parameters, it is possible to extract these from the data. This is discussed in more detail in Section 7.

5.1 The Small-r Approximation

To obtain a small-r approximation for the rate R, we treat r as a variable and write (9) as

$$R = \frac{1}{T_C} - \frac{1}{T_C} f(r\,T_C) \tag{10}$$

where the function f is defined by

$$f(x) \equiv \prod_{i=1}^{M} [3\exp(-2N_i\,x) - 2\exp(-3N_i\,x)]\,. \tag{11}$$

A series expansion about the point $r = 0$ gives

$$f(r\,T_C) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0)\,(r\,T_C)^n \tag{12}$$

where we define

$$f^{(n)}(x) \equiv \frac{d^n f(x)}{dx^n}, \qquad n = 1, 2, ...$$

$$f^{(0)}(x) \equiv f(x)\,.$$

Substituting (12) into (10) while using $f(0) = 1$ gives

$$R = -\frac{1}{T_C} \sum_{n=1}^{\infty} \frac{1}{n!} f^{(n)}(0)\,(r\,T_C)^n\,. \tag{13}$$

The next step is to calculate the derivatives of f appearing in (13). Differentiating (11) becomes increasingly laborious with each higher derivative, but a systematic method that reduces this labor is derived in the appendix. The first few derivatives are calculated in the appendix and substituting those results into (13) gives

$$R = \frac{M}{T_C}\left\{3(\mathcal{M}_2\,r\,T_C)^2 - 5(\mathcal{M}_3\,r\,T_C)^3 + \left[\frac{37}{4} - \frac{9{\mathcal{M}_2}^4}{2{\mathcal{M}_4}^4} M\right](\mathcal{M}_4\,r\,T_C)^4 + ...\right\} \tag{14}$$

where the moments $\mathcal{M}_2$, $\mathcal{M}_3$, and $\mathcal{M}_4$ are defined by

$$\mathcal{M}_2 \equiv \left[\frac{1}{M}\sum_{i=1}^{M} N_i^{\,2}\right]^{1/2}, \quad \mathcal{M}_3 \equiv \left[\frac{1}{M}\sum_{i=1}^{M} N_i^{\,3}\right]^{1/3}, \quad \mathcal{M}_4 \equiv \left[\frac{1}{M}\sum_{i=1}^{M} N_i^{\,4}\right]^{1/4}. \tag{15a}$$

The first moment $\mathcal{M}_1$ defined by

$$\mathcal{M}_1 \equiv \frac{1}{M}\sum_{i=1}^{M} N_i \tag{15b}$$

does not appear in (14) but is listed here for later discussions.

A small-r approximation is obtained by retaining only the first few terms in (14). In this approximation, it is not necessary to assign numerical values to each of the parameters N_1, N_2, …, N_M. It is sufficient to assign values to the first few moments, which become the new parameters. However, these new parameters are not entirely independent because they are constrained by inequalities. The Hölder inequality for sums (a generalization of the Cauchy-Schwarz inequality) can be used, together with (15), to show that

$$\mathcal{M}_1 \le \mathcal{M}_2 \le \mathcal{M}_3 \le \mathcal{M}_4 . \tag{16}$$

The moments are all equal if the group sizes are uniform (i.e., if $N_1 = N_2 = \ldots = N_M$), so the amount by which the moments differ provides some measure of the lack of uniformity, or spread, of the group sizes. This measure can be made more explicit by comparing, for example, M_2 to M_3. It is easy to show from (15) that

$$\mathcal{M}_3^{\,3} - \mathcal{M}_2^{\,3} = \frac{1}{M}\sum_{i=1}^{M}[(N_i + \tfrac{1}{2}\mathcal{M}_2)(N_i - \mathcal{M}_2)^2]. \tag{17}$$

Interpreting the right side of (17) as a measure of the spread in the set N_1, …, N_M, the left side is an alternate expression for this measure. A second measure of the spread is obtained from $\mathcal{M}_2$ if we know the total number of bits in the device and the total number of groups. Given this information, we also know $\mathcal{M}_1$ from (15b). It is easy to show that

$$\mathcal{M}_2^{\,2} - \mathcal{M}_1^{\,2} = \frac{1}{M}\sum_{i=1}^{M}(N_i - \mathcal{M}_1)^2 . \tag{19}$$

Interpreting the right side of (19) as a measure of the spread in the set N_1, …, N_M, the left side is an alternate expression for this measure. A third measure of the spread, which is redundant with the previous two, is obtained by comparing $\mathcal{M}_1$ to $\mathcal{M}_3$ via

$$\mathcal{M}_3^{\,3} - \mathcal{M}_1^{\,3} = \frac{1}{M}\sum_{i=1}^{M}[(N_i + 2\mathcal{M}_1)(N_i - \mathcal{M}_1)^2]. \tag{20}$$

The lowest-order small-r approximation retains only the first term in (14) so only the one moment $\mathcal{M}_2$ appears. The approximation is

$$R \approx 3M\,T_C\,(\mathcal{M}_2\,r)^2 \quad (\text{small-}r). \tag{21}$$

5.2 A More General Approximation

This section derives an approximation that is slightly more complex than (21) but has the advantage of being useful for arbitrary r. Note that there are two possible conditions that might occur. One possibility is that $\mathcal{M}_1 r T_C << 1$, and the other possibility is that $\mathcal{M}_1 r T_C$ is on the order of 1 or larger. Only the first possibility requires careful attention. The reason is as follows. Regarding $\mathcal{M}_1$ as an estimate of a typical block size, the condition $\mathcal{M}_1 r T_C \sim 1$ (or larger) states that most of the blocks will contain an error during any given cycle. With group numbers M typically in the hundreds or thousands, this condition makes it overwhelmingly probable that there will be one or more group errors (i.e., a system error) during any given cycle, so the system error rate R is one error per cycle, i.e., R is very nearly equal to $1/T_C$. Using the notation in (10), the condition is written as

$$f(r T_C) << 1 \quad \text{if } \mathcal{M}_1 r T_C \sim 1 \text{ or larger.} \tag{22}$$

Any approximation for f (even a very crude approximation) that is consistent with (22) will produce the correct R for the condition stated in (22). An approximation for f that is consistent with (22) is required to be accurate only for those conditions in which f is not negligible compared to 1, i.e., accuracy is needed only when $\mathcal{M}_1 r T_C << 1$. However, if we focus our attention to those conditions where accuracy is most needed, i.e., small-r conditions, it might appear that this analysis is just a repeat of the derivation of the small-r approximation (21). The approach used here differs from that used to derive (21) in that the approximation for f will be constructed to be consistent with (22). In contrast, (21) is not consistent with (22). Furthermore, when approximating various quantities with a partial power series in r, enough terms will be included so that accuracy is maintained as r increases up to the point where (22) applies. With accuracy available up to this point, and not needed beyond this point, the final approximation for R will be accurate for all r.

The approximation is derived by starting with an approximation for the terms in (11). The approximation used here is

$$3\exp(-2N_i\, x) - 2\exp(-3N_i\, x) \approx \exp(-3{N_i}^2 x^2 + 5{N_i}^3 x^3 - \frac{37}{4}{N_i}^4 x^4) \tag{23}$$

where $x = rT_C$. This approximation was selected partly for convenience and partly because it has two essential properties (verifications are exercises for the reader). The first property is that the error is a high-order quantity (fifth order in x to be precise). The second property is that the right side, like the left side, is strictly decreasing in x and approaches zero at large x. The first property maintains accuracy as x increases up to the point where (22) begins to apply, while the second property makes the approximation consistent with (22) so accuracy is not needed for larger x. Substituting (23) into (11) gives

$$f(x) \approx \exp\left[-3M(\mathcal{M}_2 x)^2 + 5M(\mathcal{M}_3 x)^3 - \frac{37}{4}M(\mathcal{M}_4 x)^4\right]$$

and substituting this into (10) gives

$$R \approx \frac{1}{T_C}\left\{1 - \exp\left[-3M(\mathcal{M}_2 r T_C)^2 + 5M(\mathcal{M}_3 r T_C)^3 - \frac{37}{4}M(\mathcal{M}_4 r T_C)^4\right]\right\}. \tag{24}$$

Note that (24) has the desirable property of containing only a few moments instead of the complete set of parameters $N_1, \ldots, N_M$. Also, a numerical comparison for a specific example (in which $N_i = 200$ for each i, $M = 48{,}000$, and $T_C = 5$ ms) found that the approximation (24) agrees with the exact equation (9) to 5 or 6 digits (or better) for all $r > 0$; when adequate arithmetic precision is used to evaluate both equations. However, (24) has the undesirable property of subtracting nearly equal numbers (the two terms in the curly brackets) when r is small. This is a numerical problem when arithmetic precision is limited. A solution to this problem is to note that (24), when evaluated with adequate arithmetic precision, is virtually indistinguishable from the small-r approximation (21) when the latter approximation predicts $RT_C < 10^{-5}$. The final approximation then becomes

$$R \approx \begin{cases} 3M\,T_C\,(\mathcal{M}_2\,r)^2 & \text{if numerical precision is limited and } 3M\,(\mathcal{M}_2\,r\,T_C)^2 < 10^{-5} \\ \text{right side of (24) otherwise.} \end{cases} \tag{25}$$

6. Some Questions that are Easily Answered via an Expansion in Moments

Suppose all device parameters are known so we have a choice between (9) and (14). However, there are certain questions whose answers are much more obvious from (14) than from (9). Some examples are given below. To be more specific, (14) will be used to compare error rates produced by different device architectures. It will be seen that (14) easily derives predictions that are not at all obvious from a casual inspection of (9).

6.1 Comparison Between Counting Conventions

The first comparison considered compares different counting conventions. Recall that simultaneous errors in multiple groups are being counted as one system error. An alternate counting convention counts multiple group errors as multiple system errors, so there can be more than one system error in a cycle. The latter convention is less useful to a system designer but leads to a simpler equation for the error rate. Note that the two conventions will produce the same system error rates in the small-r limit (because the probability of two or more group errors becomes much smaller than the probability of one group error), but it is interesting to determine the rates at which the two conventions begin to depart from each other. The system error rate for the alternate (multiple error) counting convention is denoted R^* and is the sum of group error rates summed over all groups. The error rate for the i^{th} group is denoted R_i and is given by

$$R_i = \frac{1}{T_C} P(E_i)$$

so R^* is given by

$$R^* = \frac{1}{T_C} \sum_{i=1}^{M} P(E_i).$$

Combining this with (3) and (6) and expressing each exponential function in (3) as a power series gives

$$R^* = \frac{M}{T_C}\left\{3(\mathcal{M}_2\, r\, T_C)^2 - 5(\mathcal{M}_3\, r\, T_C)^3 + \frac{19}{4}(\mathcal{M}_4\, r\, T_C)^4 + \ldots\right\}. \tag{26}$$

Note that the two rates in (14) and (26) differ only in the fourth and higher power terms. The conclusion is that the two rates are indistinguishable until r is large enough for the fourth or higher power terms to become significant contributions.

6.2 Uniform Versus Non-Uniform Group Sizes

The next comparison investigates the benefit of using uniform group sizes. Suppose a device designer is constrained to use a fixed number of bits distributed among a fixed number of groups, but there is flexibility in the sizes of the groups subject to those constraints, i.e., the groups can be of equal size, or bits can be taken from one group and put into another so that some groups are larger than others. The objective is to compare system error rates for the different possibilities in order to determine the optimum group construction. We will answer this question when r is sufficiently small so that (14) reduces to (21), and using (19) gives

$$R = 3M\, T_C \left[\mathcal{M}_1{}^2 + \frac{1}{M}\sum_{i=1}^{M}(N_i - \mathcal{M}_1)^2\right] r^2 .$$

The mean number of bits $\mathcal{M}_1$ is the same for all cases being compared, so different cases differ in the summation term. The construction that produces the smallest R is the one in which the summation term vanishes, i.e., all groups are of the same size. This means that if we start with equal size groups and then take bits out of one group (making it smaller, which tends to decrease R) and put them into another group (making it larger, which tends to increase R), the tendency for R to increase outweighs the tendency to decrease. Note that this conclusion is not at all obvious from a casual inspection of (9).

6.3 TMR Versus Hamming Code

The last comparison is between TMR and the Hamming code. Again we consider the small-r limit so the rate for the TMR arrangement is given by (21). We also assume that the TMR arrangement uses uniform group sizes so $N_1 = N_2 = \ldots = N_M = N_{B/B}$, where the notation $N_{B/B}$ is a reminder that this is the number of bits per block. Note that $\mathcal{M}_2 = N_{B/B}$, so the rate for the TMR arrangement becomes

$$[R]_{TMR} = 3M\, N_{B/B}{}^2\, T_C\, r^2 .$$

An earlier analysis found that the system error rate for the Hamming code device, in the same small-r limit, is given by

$$[R]_{Hamming} = \tfrac{1}{2} N_W\, N_{B/W}\, (N_{B/W} - 1)\, T_C\, r^2$$

where N_W is the number of EDAC words in the device and $N_{B/W}$ is the number of bits per word (including check bits). To make the comparison fair, we assume that both devices have the same total number of bits, denoted N_{Total}. For the TMR device the total number of bits is given by

$$N_{Total} = 3M\, N_{B/B}$$

and for the Hamming code device it is given by

$$N_{Total} = N_{B/W}\ N_W$$

so the ratio of the rates is given by

$$\frac{[R]_{TMR}}{[R]_{Hamming}} = \frac{2\,N_{B/B}}{N_{B/W} - 1} \qquad \text{(same } N_{Total} \text{ for both).}$$

The number of bits per block in the TMR device is one-third the number of bits per group, denoted $N_{B/G}$, so an alternate equation is

$$\frac{[R]_{TMR}}{[R]_{Hamming}} = \frac{2}{3}\frac{N_{B/G}}{N_{B/W} - 1} \qquad \text{(same } N_{Total} \text{ for both).}$$

The TMR device will have the smaller rate if the groups are small enough (and correspondingly numerous to maintain the total number of bits) to make the right side less than unity.

7. Interpretation of Test Data Obtained from a Particle Accelerator

If all of the device parameters (r, T_C, M, N_1, …, N_M) are given inputs, the system error rate R can be calculated directly from (9). However, suppose that these parameters are not all given. Instead, the given information consists of test data (measured error rates) produced at a particle accelerator. One possible goal is to extract information regarding device architecture from these data. Another possible goal is to estimate the system error rate in a given space environment. In either case, the method used to obtain and present test data is the same, and is discussed in Section 7.1. The role that the theory plays in data interpretation is discussed in Section 7.2, and some example applications are given in Section 7.3.

7.1 Obtaining and Presenting Test Data

Measuring single-event effect cross sections is a familiar procedure, so we begin the discussion by using this language. There are two sets of tests, one with TMR turned off, and the other with TMR turned on. With TMR turned off, the first set of tests vary the ion linear energy transfer (LET) L and measures the per-bit cross section, denoted $\sigma_{per\text{-}bit}(L)$, for each LET. For the second set of tests, TMR is turned on and a convenient LET is selected, where "convenient" means that adequate counting statistics can be obtained from flux levels available at the accelerator and using reasonable beam run times (tests become very expensive when beam run times are more than a few minutes per run). In this set of tests, the flux f is varied from one run to the next. For each run, the system-error cross section (per device and defined in the usual way, i.e., counts divided by fluence), denoted $\sigma_{SYS}(f)$, is measured and the flux is recorded. This produces a plot of the system-error cross section as a function of flux. This plot is then converted into a plot of R versus r by multiplying the vertical coordinate $\sigma_{SYS}(f)$ by f to obtain R, and by multiplying the horizontal coordinate f by $\sigma_{per\text{-}bit}(L)$ (where L is the LET used during the second set of tests) to obtain r. The final result is an experimental determination of R as a function of r.

7.2 Using Theory to Augment Test Data

With an experimental determination of R versus r available, it is reasonable to ask what the purpose is of all of the theory derived in this paper. It would appear that we could use standard rate calculation methods to estimate r for a given space environment, using the experimentally measured cross section versus LET curve for raw bit flips, and then combine this estimate with the experimentally measured plot of R versus r to obtain an estimate of the system error rate R in that environment. It might appear that all required information can be obtained experimentally. There are several reasons why the theory is useful:

(a) *Test Data Extrapolation:* Test data extrapolation is probably the most important motivation for the theory because this will have to be done each time a rate estimate is needed in a space environment. Additional discussion below will explain why there are practical constraints regarding the flux levels that can be used during a particle-accelerator test, but for now it is enough to know that test flux levels will be much larger than flux levels encountered in space environments, i.e., r values produced in a test are much larger than produced by a space environment. The R versus r plot will have to be extrapolated to the smaller (by multiple decades) values of r produced in space. A theoretical model is needed for a multiple-decade extrapolation. There is an important caveat. An extrapolation cannot be trusted unless it is known (e.g., from electrical tests) that the device has the expected functionality. Otherwise, the device might belong to item (b) below.

(b) *Device Verification:* It is not always clear whether a real device has met expectations. Not all device elements are triplicated, i.e., not all elements are protected by TMR. Perhaps some portion of the device, not protected by TMR, will make the dominant contribution to the error rate. Or, perhaps a portion that was thought to be protected by TMR really isn't due to a design or fabrication flaw. A third possibility is that a bad device layout results in a system error when multiple-bit upsets are produced by a single ion hit. By comparing an experimentally measured R versus r plot to the ideal (i.e., theoretical) prediction, a device designer or manufacturer can determine whether the ideal limit has been reached.

(c) *Device Parameter Extraction:* One possible goal is to extract information regarding device architecture (e.g., the numbers of bits in the groups). Some information (e.g., the root-mean-square) can be obtained by fitting theoretically derived curves to the test data.

(d) *Dealing with Scatter:* Cross-section data sometimes contain enough scatter to obscure the underlying curve that the data would conform to if there were no scatter. When attempting to fit such data with a curve, but without any theoretical guidance, it is not clear whether the curve should be a straight line in a log-log plot, a straight line in a log-linear plot, or something else. Theoretical guidance can help to determine the type of curve that should be used.

Item (a) above requires additional discussion. It might appear that flux levels and/or ion LET can be varied as needed to make r as small as desired or as large as desired, so an experimental plot of R versus r could be made to extend from the small-r regime to the large-r regime. However, a practical consideration limits the plotted range so that the regime that is sampled is more strongly controlled by T_C than by user preference. This practical consideration

is the need to obtain adequate counting statistics while avoiding the expense associated with excessive beam time. To obtain adequate counting statistics, a beam run should produce at least several tens of system errors. Let us arbitrarily select 20 system errors as a base-line number. To avoid excessive beam time, the test run time should be relatively short. Let us arbitrarily select 100 seconds as a base-line number. To produce these base-line numbers, the flux must be adjusted to be whatever is needed to produce a system error rate R that is 0.2/s. There is some flexibility (e.g., the run time could be an hour), but a typical value of R will be within one or two orders of magnitude of this base-line number. However, this value of R might belong to the small-r regime, or it might belong to the large-r regime, depending on how this value compares to $1/T_C$ (because the small-r regime applies when $R << 1/T_C$, while the large-r regime applies when $R \sim 1/T_C$). Because the user has only limited control over R, the comparison between R and $1/T_C$ is most strongly controlled by T_C. For example, if T_C is one millisecond, the experimental data will probably be sampling the small-r regime. If T_C is several seconds or larger, the experimental data will probably be sampling the large-r regime. For any case in which the data do not sample both regimes, theory is needed to extrapolate the data from one regime into the other.

7.3 Examples

A Xilinx field-programmable gate array (FPGA) was configured in several ways, identified here as "BRSCRUB", "Multipliers", and "Counters". Accelerator test data were obtained by Carl Carmichael (Xilinx), Gary Swift (JPL), Greg Allen (JPL), and Sana Rezgui (Xilinx), using the method in Section 7.1. Data interpretation (e.g., distinguishing TMR errors from other types of errors, such as single-event functionality interrupts (SEFIs)) was performed by Jeffrey George (Aerospace), Gary Swift (JPL), Carl Carmichael (Xilinx), and Tilan Langley (JPL). Test data presented in the format discussed in Section 7.1 are shown as the points in Figures 2 through 4. The curves are discussed below.

To construct fits to the data, it is desirable to obtain as much device-related information as possible in order to minimize the number of device parameters that must be treated as adjustable fitting parameters. The same individuals that participated in data interpretation have provided the following supplemental information:

- For the BRSCRUB, the cycle time (T_C) is 2 ms, and the number of groups (M) is 48,000. Also, the group sizes are uniform, so $\mathcal{M}_2 = \mathcal{M}_3 = \mathcal{M}_4$. This leaves only one adjustable fitting parameter $\mathcal{M}_2$.
- For the Counters, the cycle time (T_C) is 0.266 s, and the number of groups (M) is 8,224. Also, the group sizes are nearly uniform in that 8,192 of the groups are of the same size, although the remaining 32 groups might be of different sizes. It is assumed that the non-uniformity produced by the latter groups is not large enough to invalidate the approximation $\mathcal{M}_2 \approx \mathcal{M}_3 \approx \mathcal{M}_4$, so there is only one adjustable fitting parameter $\mathcal{M}_2$.
- For the Multipliers, the cycle time (T_C) is 0.266 s, and the number of groups (M) is 900. Also, the group sizes are nearly uniform in that 864 of the groups are of the same size, although the remaining 36 groups might be of different sizes. It is assumed that the non-uniformity produced by the latter groups is not large enough to invalidate the approximation $\mathcal{M}_2 \approx \mathcal{M}_3 \approx \mathcal{M}_4$, so there is only one adjustable fitting parameter $\mathcal{M}_2$.

In each case there is only one adjustable fitting parameter $\mathcal{M}_2$. Assigning values to this parameter to obtain what appears to be a best fit to the data produces the curves in Figures 2 through 4 (parameters are listed in the figure captions). In each case, the curve produced by the general approximation (24) is indistinguishable (error is less than the line thickness for all r) from the curve produced by the exact equation (9). Although incidental, it is interesting to note that the small-r approximation agrees well with the exact equation until R is within an order of magnitude of its limiting value.

Note that a suitable choice of fitting parameters gives good agreement (within experimental scatter) between the fit and data for the BRSCRUB in Figure 2. In contrast, the three points for the Multipliers in Figure 4 suggest that a different curve may apply. This might be an illusion produced by experimental scatter (with only three points we cannot distinguish scatter from a systematic trend). Another possibility is a flaw in the device fabrication or design such as discussed in Item (b) in Section 7.2. Additional data points (enough so that we can distinguish scatter from a systematic trend) would be needed to resolve this issue, so results for the Multipliers are inconclusive at this time. However, the comparison between data and theory does at least alert us to the fact that the Multipliers should be investigated for possible design or fabrication flaws.

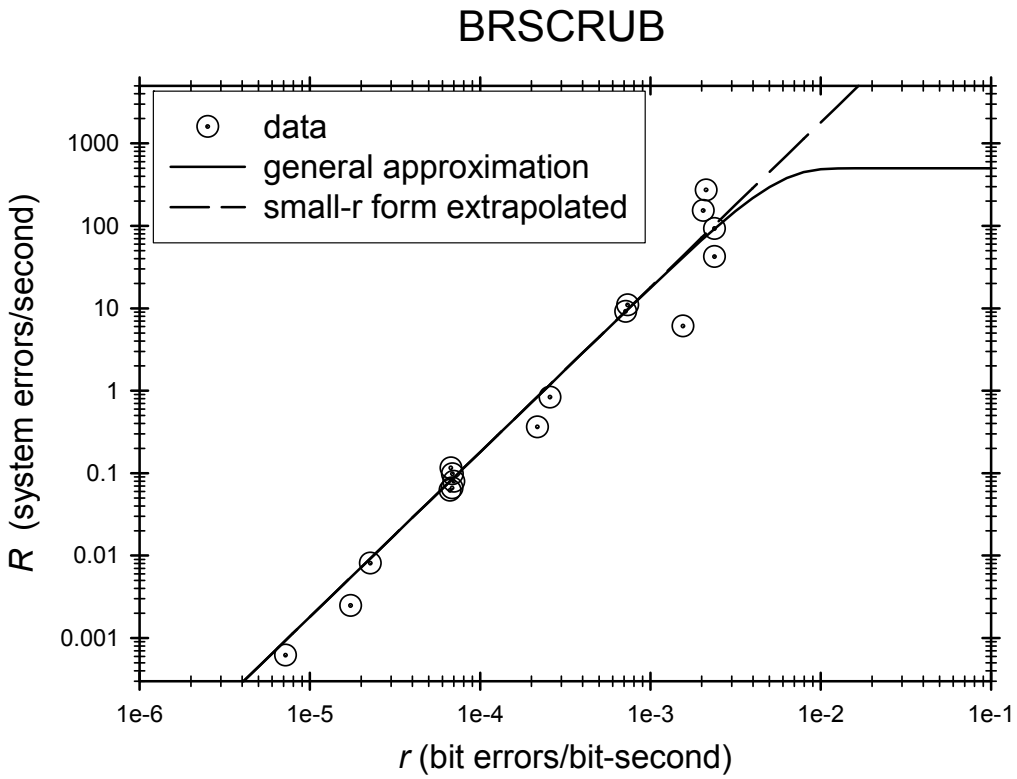


Figure 2. Data and Fits for the BRSCRUB

The fits used T_C = 2 ms, M = 48000, and $\mathcal{M}_2 = \mathcal{M}_3 = \mathcal{M}_4$ = 250. Note that T_C and M were given, and it was given that $\mathcal{M}_2 = \mathcal{M}_3 = \mathcal{M}_4$, so $\mathcal{M}_2$ is the only adjustable fitting parameter. The curve produced by the general approximation (24) is indistinguishable (error is less than the line thickness for all r) from the curve produced by the exact equation (9). Also, the small-r approximation agrees well with the exact equation until R is within an order of magnitude of its limiting value.

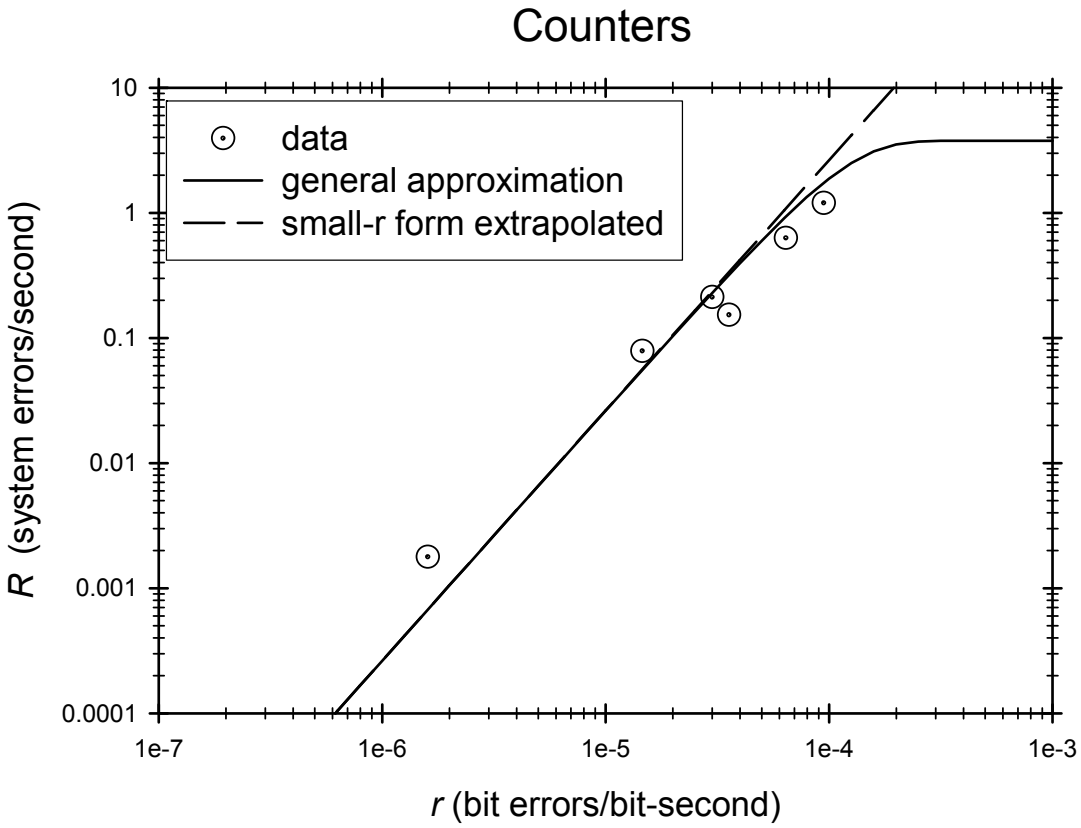


Figure 3. Data and Fits for the Counters

The fits used $T_C = 0.266$ s, $M = 8224$, and $\mathcal{M}_2 = \mathcal{M}_3 = \mathcal{M}_4 = 200$. Note that T_C and M were given, and it was assumed that $\mathcal{M}_2 \approx \mathcal{M}_3 \approx \mathcal{M}_4$, so $\mathcal{M}_2$ is the only adjustable fitting parameter. The curve produced by the general approximation (24) is indistinguishable (error is less than the line thickness for all r) from the curve produced by the exact equation (9). Also, the small-r approximation agrees well with the exact equation until R is within an order of magnitude of its limiting value.

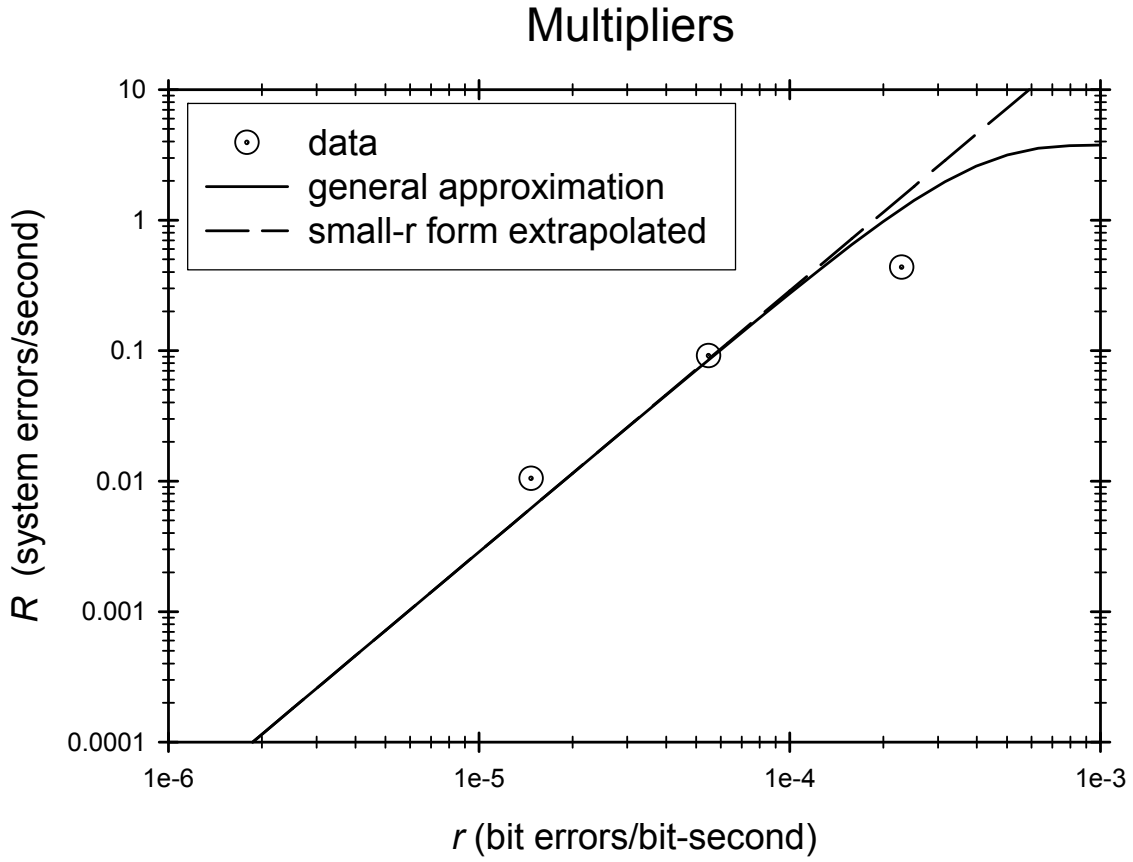


Figure 4. Data and Fits for the Multipliers

The fits used $T_C = 0.266$ s, $M = 900$, and $\mathcal{M}_2 = \mathcal{M}_3 = \mathcal{M}_4 = 200$. Note that T_C and M were given, and it was assumed that $\mathcal{M}_2 \approx \mathcal{M}_3 \approx \mathcal{M}_4$, so $\mathcal{M}_2$ is the only adjustable fitting parameter. The curve produced by the general approximation (24) is indistinguishable (error is less than the line thickness for all r) from the curve produced by the exact equation (9). Also, the small-r approximation agrees well with the exact equation until R is within an order of magnitude of its limiting value.

Appendix: Evaluation of the Derivatives in (13)

A systematic method for evaluating the derivatives in (13) is obtained by first defining f_i by

$$f_i(x) \equiv 3\exp(-2N_i\, x) - 2\exp(-3N_i\, x) \tag{A1}$$

so (11) becomes

$$f(x) = \prod_{i=1}^{M} f_i(x)\,. \tag{A2}$$

Now define g and g_i by

$$g(x) \equiv \ln(f(x)), \qquad g_i(x) \equiv \ln(f_i(x)) \tag{A3}$$

so that (A2) becomes

$$g(x) = \sum_{i=1}^{M} g_i(x)\,. \tag{A4}$$

It can be shown by induction that if two functions f and g are related by (A3), then the derivatives are related by

$$f^{(n)}(x) = (n-1)!\sum_{i=1}^{n} \frac{f^{(n-i)}(x)\, g^{(i)}(x)}{(i-1)!\,(n-i)!}, \qquad n = 1, 2, \ldots \tag{A5}$$

which is a recursion formula that solves for the highest derivative of f in terms of derivatives of g and lower derivatives of f. A similar equation relates derivatives of f_i to derivatives of g_i, but it is convenient to write this equation so that the highest derivative of g_i appears on the left side. The equation is

$$g_i{}^{(1)}(x) = \frac{f_i{}^{(1)}(x)}{f_i(x)} \tag{A6a}$$

$$g_i{}^{(n)}(x) = \frac{f_i{}^{(n)}(x)}{f_i(x)} - \frac{(n-1)!}{f_i(x)}\sum_{j=1}^{n-1} \frac{f_i{}^{(n-j)}(x)\, g_i{}^{(j)}(x)}{(j-1)!\,(n-j)!}, \qquad n = 2, 3, \ldots \tag{A6b}$$

There are now enough equations to calculate all derivatives when the derivatives of f_i are given inputs. For the case considered here, f_i is given by (A1) so the derivatives are

$$f_i{}^{(n)}(x) = 3(-2N_i)^n \exp(-2N_i\, x) - 2(-3N_i)^n \exp(-3N_i\, x), \qquad n = 0, 1, \ldots \tag{A7}$$

The advantage of this method is that all other derivatives are calculated from the derivatives in (A7) using algebraic (as opposed to differential) operations. This means that the above derivatives can be evaluated at $x = 0$ (which greatly simplifies the expressions) in order to obtain the other derivatives also evaluated at $x = 0$. From here on we shorten the notation by omitting the argument x with the understanding that all functions from this point on are evaluated at $x = 0$. Evaluating (A7) at $x = 0$ and explicitly listing the first five terms gives

$$f_i = f_i^{(0)} = 1, \quad f_i^{(1)} = 0, \quad f_i^{(2)} = -6N_i{}^2, \quad f_i^{(3)} = 30N_i{}^3, \quad f_i^{(4)} = -114N_i{}^4. \tag{A8}$$

Explicitly listing the first few equations in (A6) while using the fact that $f_i = 1$ gives

$$\left.\begin{aligned} g_i^{(1)} &= f_i^{(1)}, \qquad g_i^{(2)} = f_i^{(2)} - f_i^{(1)}\, g_i^{(1)} \\ g_i^{(3)} &= f_i^{(3)} - f_i^{(2)}\, g_i^{(1)} - 2f_i^{(1)}\, g_i^{(2)} \\ g_i^{(4)} &= f_i^{(4)} - f_i^{(3)}\, g_i^{(1)} - 3f_i^{(2)}\, g_i^{(2)} - 3f_i^{(1)}\, g_i^{(3)}. \end{aligned}\right\} \tag{A9}$$

Substituting (A8) into (A9) solves for the derivatives of g_i. Putting these results into (A4) gives

$$g^{(1)} = 0, \quad g^{(2)} = -6\sum_{i=1}^{M} N_i{}^2, \quad g^{(3)} = 30\sum_{i=1}^{M} N_i{}^3, \quad g^{(4)} = -222\sum_{i=1}^{M} N_i{}^4. \tag{A10}$$

Explicitly listing the first few equations in (A5) while using the fact that $f = 1$ gives

$$\left.\begin{aligned} f^{(1)} &= g^{(1)}, \qquad f^{(2)} = f^{(1)}\, g^{(1)} + g^{(2)} \\ f^{(3)} &= f^{(2)}\, g^{(1)} + 2f^{(1)}\, g^{(2)} + g^{(3)} \\ f^{(4)} &= f^{(3)}\, g^{(1)} + 3f^{(2)}\, g^{(2)} + 3f^{(1)}\, g^{(3)} + g^{(4)}. \end{aligned}\right\} \tag{A11}$$

Substituting (A10) into (A11) solves for the derivatives of f evaluated at $x = 0$, and the results are

$$f^{(1)} = 0, \qquad f^{(2)} = -6\sum_{i=1}^{M} N_i{}^2, \qquad f^{(3)} = 30\sum_{i=1}^{M} N_i{}^3$$

$$f^{(4)} = 6\left\{18\left[\sum_{i=1}^{M} N_i{}^2\right]^2 - 37\sum_{i=1}^{M} N_i{}^4\right\}.$$